LAPORAN RESMI

PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK

OVERLOADING DAN OVERRIDING

Fadilah Fahrul Hardiansyah S.ST., M. Kom

Ratri Maria Manik

3121600039

D4 TEKNIK INFORMATIKA – B

PROGRAM STUDI TEKNIK INFORMATIKA

POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

TA 2022/2023

A. TUGAS PENDAHULUAN

1. Jelaskan perbedaan overloading method dan overloading constructor!
   a. Overloading terhadap method
      Aturan pendeklarasian overloading terhadap method
      - Nama method harus sama
      - Daftar parameter harus berbeda
      - Return type boleh sama, juga boleh berbeda

      Contoh overloading method :

```
public int isiNilai(int nilai){
        return nilai * nilai;
}

public float isiNilai(float nilai){
        return nilai * nilai;
}
```

   b. Overloading terhadap konstruktor
      Suatu mekanisme pembuatan konstruktor yang memiliki bentuk lebih dari satu. Dalam hal ini pembeda antara satu konstruktor dengan konstruktor yang lain berupa jumlah parameter atau tipe parameter.

      Contoh overloading konstruktor :

```
public buku(String pengarang, String judul){
        this.pengarang = pengarang;
        this.judul = judul;
}

public buku(){
}
```

2. Apakah overloading bisa terjadi antara superclass dan cubclass? Jika bisa, berikan contohnya!
   Ya, overloading dapat terjadi antara superclass dan subclass apabila memenuhi ketiga syarat overload. Contoh terjadinya overloading antara superclass dan subclass yaitu memiliki method yang sama, dengan daftar parameter yang berbeda.

3. Apakah overloading constructor bisa terjadi antara superclass dan subclass? Jika bisa berikan contohnya!
   Overloading constructor tidak bisa terjadi antara superclass dan subclass karena constructor memiliki nama yang sama dengan nama classnya, sedangkan nama class antara superclass dan subclass tidak boleh sama.

B. LATIHAN

1. Melakukan overriding, overloading, dan overloading constructor
   Perbaiki class diagram berikut dengan mengimplementasikan enkapsulasi, kemudian implementasikan menjadi kode program.
   a. Class User.java

```
public class User{
        private String id;
        private String phone;
        private String username;
        private String password;

        public User(){
        }

        public User(String username, String password){
                this.username = username;
                this.password = password;
        }

        public void setUsername(String username){
                this.username = username;
        }

        public void setPassword(String password){
                this.password = password;
        }

        public void setPhone(String phone){
                this.phone = phone;
        }

        public boolean login(String username, String password){
                if(this.username == username && this.password ==
password){
                        return true;
                }
                else{
                        return false;
                }
        }

}
```

b. Class Dosen.java

```
public class Dosen extends User{
	private String pangkat;
	private String golongan;
	private String key;

	public Dosen(){}

	public Dosen(String username, String password){
		super(username, password);
	}

	public void setKey(String key){
		this.key = key;
	}

	public boolean login(String username, String password){
		if(super.login(username, password)){
			return true;
		}
		else{
			return false;
		}
	}

	public boolean login(String username, String password, String key){
		if(super.login(username, password) && this.key == key){
			return true;
		}
		else{
			return false;
		}
	}
}
```

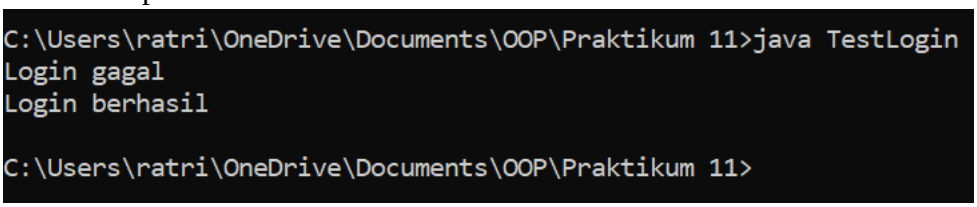
c. Class Mahasiswa.java

```
public class Mahasiswa extends User{
	private Dosen dosenWali;
	private String kelas;

	public Mahasiswa(){}

	public Mahasiswa(String username, String password){
		super(username, password);
	}
}
```

d. Class TestLogin.java

```
public class TestLogin{
	public static void main(String arguments[]){
		Mahasiswa fadilah = new Mahasiswa("Fadilah", "123456");
		boolean result = fadilah.login("083853501388", "123456");
		if(result)
			System.out.println("Login berhasil");
		else
			System.out.println("Login gagal");

		Dosen fahrul = new Dosen("fahrul","123456", "19990129");
		result = fahrul.login("fahrul", "123456", "19990129");
		if(result)
			System.out.println("Login berhasil");
		else
			System.out.println("Login gagal");
	}
}
```

e. Hasil Kompile



Analisa :

Pada program di atas menggunakan metode enkapsulasi, sehingga terdapat metode setter, namun penggunaan terhadap metode getter tidak digunakan karena tidak diperlukan. Digunakan **super()** untuk menjalankan konstruktor parentnya

C. TUGAS

Modifikasi program pada soal Latihan dengan ,enghilangkan method setter & getter pada semua class, tapi setiap variabel tetap harus menggunakan modifier private

a. Class User.java

```
public class User{
	private String id;
	private String phone;
	private String username;
	private String password;

	public User(){
	}

	public User(String username, String password){
		this.username = username;
		this.password = password;
	}

	public boolean login(String username, String password){
		if(this.username == username && this.password == password){
			return true;
		}
		else{
			return false;
		}
	}

}
```

b. Class Dosen.java

```
public class Dosen extends User{
	private String pangkat;
	private String golongan;
	private String key;

	public Dosen(){}

	public Dosen(String username, String password, String key){
		super(username, password);
		this.key = key;
	}

	public boolean login(String username, String password){
		if(super.login(username, password)){
			return true;
		}
```

```
                else{
                        return false;
                }
        }

        public boolean login(String username, String password, String key){
                if(super.login(username, password) && this.key == key){
                        return true;
                }
                else{
                        return false;
                }
        }
}
```

c. Class TestLogin.java

```
public class TestLogin{
        public static void main(String arguments[]){
                Mahasiswa fadilah = new Mahasiswa("Fadilah", "123456");
                boolean result = fadilah.login("083853501388", "123456");
                if(result)
                        System.out.println("Login berhasil");
                else
                        System.out.println("Login gagal");

                Dosen fahrul = new Dosen("fahrul","123456", "19990129");
                result = fahrul.login("fahrul", "123456", "19990129");
                if(result)
                        System.out.println("Login berhasil");
                else
                        System.out.println("Login gagal");
        }
}
```

d. Hasil kompile

```
C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 11>javac TestLogin.java

C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 11>java TestLogin
Login gagal
Login berhasil

C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 11>
```

Analisa :

Menghilangkan metode setter and getter pada setiap kelas, program diatas yang menggunakan metode tersebut semua class kecuali class Mahasiswa kemudian apabila program dikompile, maka tampilannya akan seperti diatas.